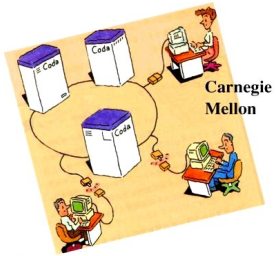
Mobile Data Access

David B. Johnson

School of Computer Science
Carnegie Mellon University

http://www.monarch.cs.cmu.edu/glomo.html
dbj@cs.cmu.edu

CMU Monarch Project



Carnegie
Mellon

Mobile Data Access

Three interrelated parts of Mobile Data Access GloMo project

Networking level:

- ***Monarch*** adaptive mobile networking protocols:
  - Routing, and also data link and transport layers
  - Also active contribution to Internet standards through IETF

Middleware/application level:

- ***Coda*** application-transparent adaptation:
  - File system performs all adaptation and resource management
  - Important for legacy applications (no source code, etc.)
- ***Odyssey*** application-aware adaptation:
  - Adaptation is collaboration between system and application
  - Flexibility, but with system control over scarce resources

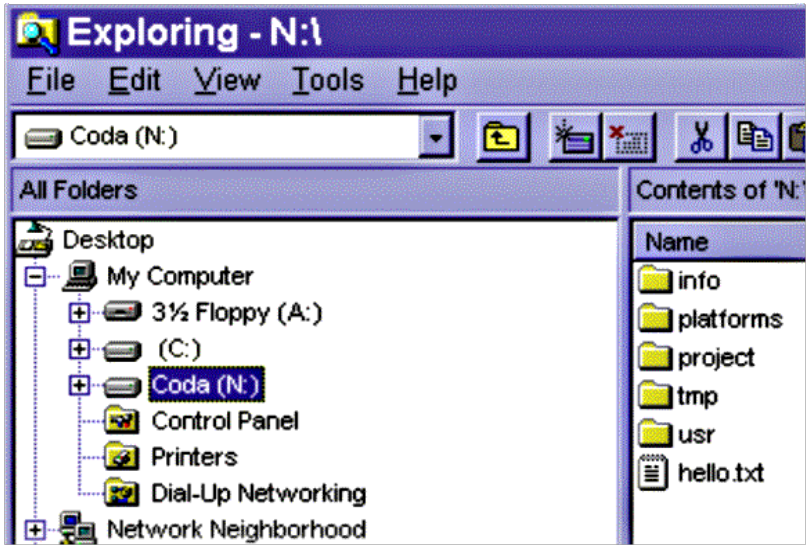
Monarch can be layered under Coda or Odyssey or other applications

Coda and Odyssey

Coda and Odyssey work under GloMo Mobile Data Access is complete

Recent results on Coda:

- Coda client ported to Windows 95 and Windows 98
- Coda server ported to Windows NT
- Also read-only NT Coda client
- Unix and Windows versions built from a single source base



Recent results on Odyssey:

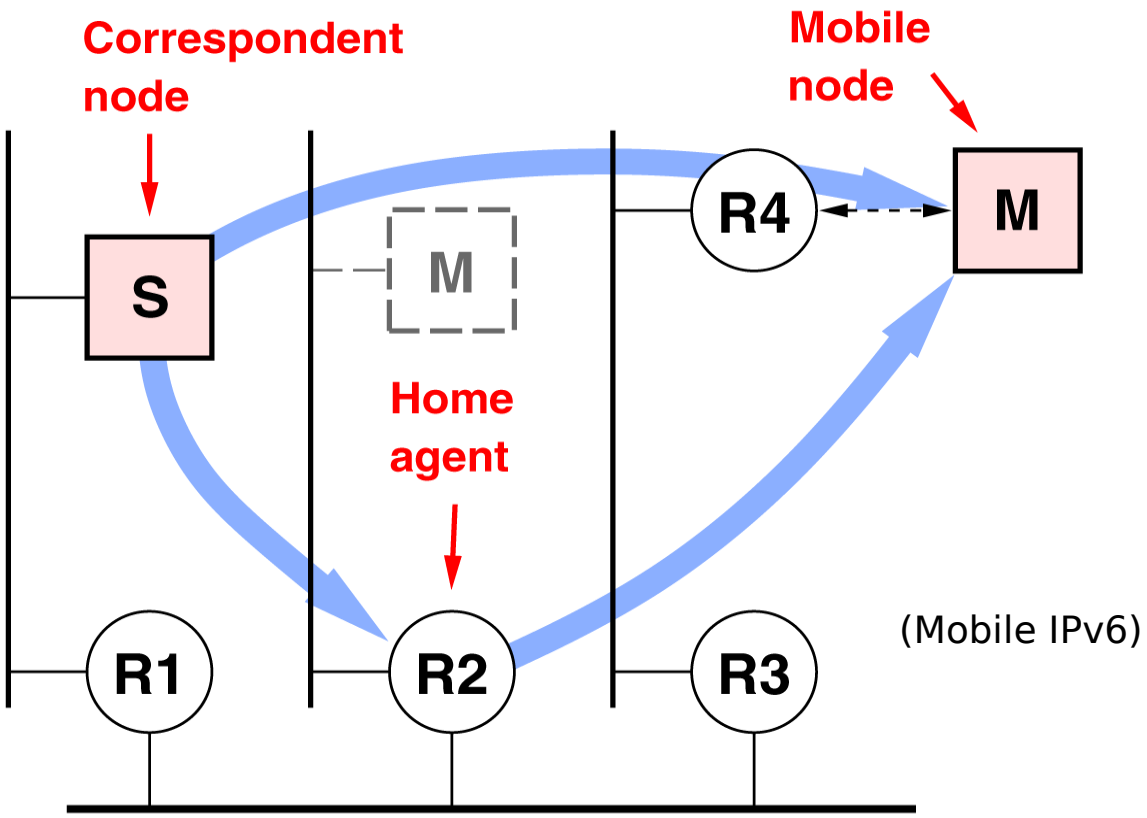
- Designed and implemented goal-directed energy management
- Developed PowerScope: A tool for profiling energy usage of applications
- Maps power consumption to program structure, like CPU time profiling
- Example: Reduced energy use of Odyssey video application by 46%

Monarch

Anytime, anywhere best use of best available connectivity:

- All mobility may be transparent to protocols and applications
- Network API for adaptation by/from mobile-aware higher layers
- Integrated multi-hop wireless ad hoc network routing and Mobile IP
- Mobility and routing between heterogeneous types of networks
- Support for multicast and adaptive QoS
- Improvements for good TCP performance
- Complete simulation with models for all protocols and layers
- Complete implementation in experimental testbed for experience, measurement, and simulation validation
- Compatible with (and part of) Internet Standards

Mobile IP for IPv4 and IPv6



Mobile IP for IPv4 is already an IETF Internet standard (RFC 2002)

Mobile IP for IPv6 final draft version of specification submitted Feb 2000

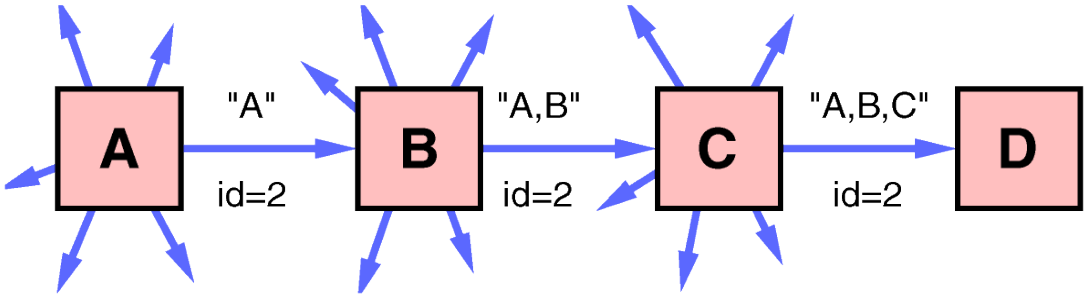
Route Optimization for Mobile IPv4 standardization later in FY00

Multi-Hop Wireless Ad Hoc Networking

Our ***Dynamic Source Routing*** (DSR) protocol is simple and efficient

Unique properties of the protocol:

- Eliminates ***all*** periodic routing or other packets
- Nodes ***ignore*** all topology changes not affecting them
- Overhead scales ***automatically*** as movement increases
- ***Zero*** overhead when stationary and found routes already
- Can support ***unidirectional*** links and ***asymmetric*** routes
- Integrated with Mobile IP for nodes ***visiting*** an ad hoc network



Under consideration in IETF MANET Working Group

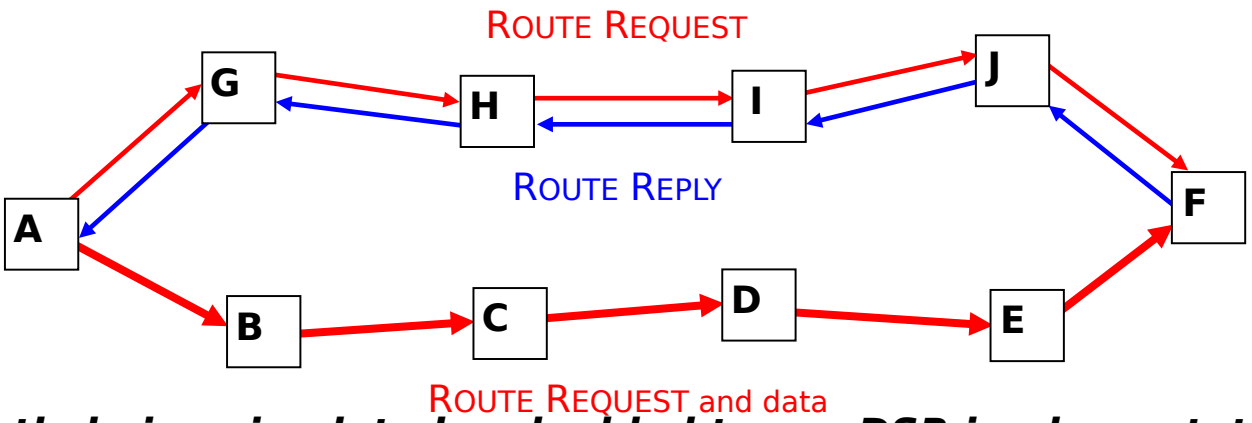
DSR Adaptive QoS Support

New path-state maintenance mechanism being added to DSR:

- Removes source route header overhead from most data packets
- Allows QoS reservation setup and feedback to adaptive applications
- Described in October 1999 Internet-Draft (draft-ietf-manet-dsr-03.txt)

A natural extension to DSR Route Request/Reply/Error mechanisms:

- Route Request carries reservation request, only propagated by nodes that can meet minimum level requested
- Changes in network requiring QoS renegotiation send Route Error



Currently being simulated and added to our DSR implementation

Improvements to DSR Route Caching

Any ***on-demand*** routing protocol for ad hoc networks must use caching:

- Avoids overhead of Route Discovery before every data packet
- But there is a cost to keeping a cache entry “too long”
- And there is a cost to keeping a cache entry “not long enough”

Experimenting with adaptive cache timeout and improved data structures:

- Cache individual links, not whole routes, similar to link state routing
- Simple passage of time is not what makes a cached route go bad
- Instead, track mobility of each node by tracking lifetime of links in cache
- Assign timeout to new cached link based on node endpoints

Preliminary results with new DSR caching:

- Packet delivery ratio was 98%, now 99% for constant motion (avg 10 M/s)
- Routing packet overhead reduced by a factor of up to 2
- Average packet delivery latency equal or somewhat improved
- Path optimality in number of hops roughly unchanged

Emulating Ad Hoc Networks

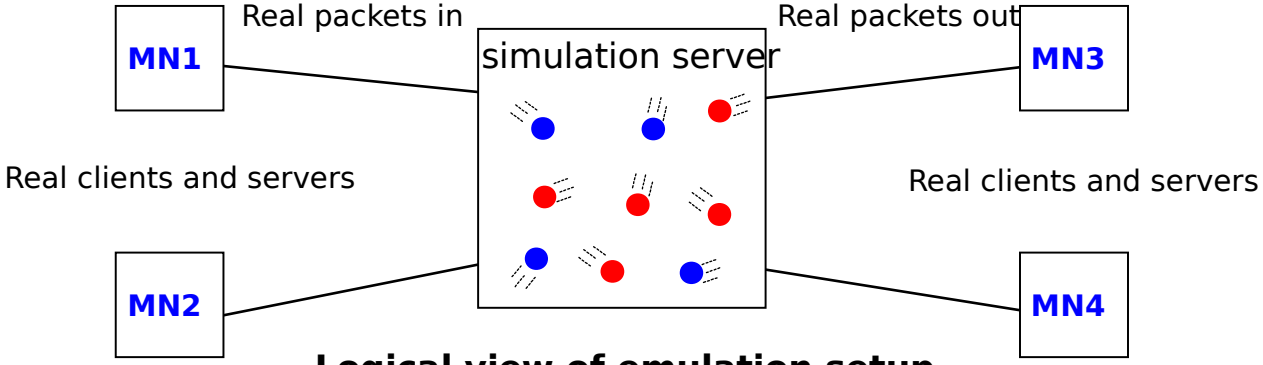
Evaluating ***real systems*** with just simulation is not practical:

- Difficult to model real applications inside the simulator
- Real systems typically are significantly performance tuned

Developed ***network emulation*** tool to address this:

- Using the ***real*** application implementation and ***real*** user patterns
- ***Only*** the network environment simulated
- Real system experiences life as if running on real ad hoc network

Demonstrated 2 real Coda clients and servers on DSR ad hoc network



Logical view of emulation setup

Evaluating Emulator Scalability

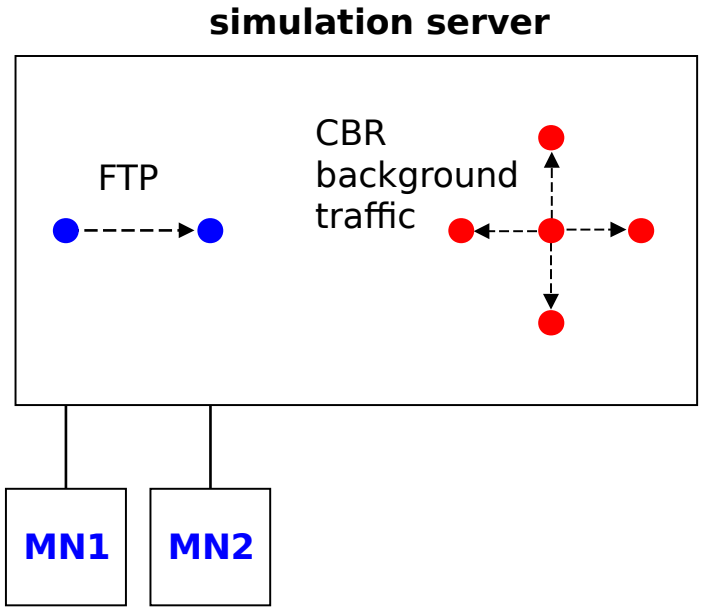
Limitations on emulation:

- System effects: scheduling delays, potential buffer overruns, ...
- Additional latency of transferring packets to/from simulation server
- Only so much one CPU can do as the simulation server: ***fundamental limit is the real-time requirement***

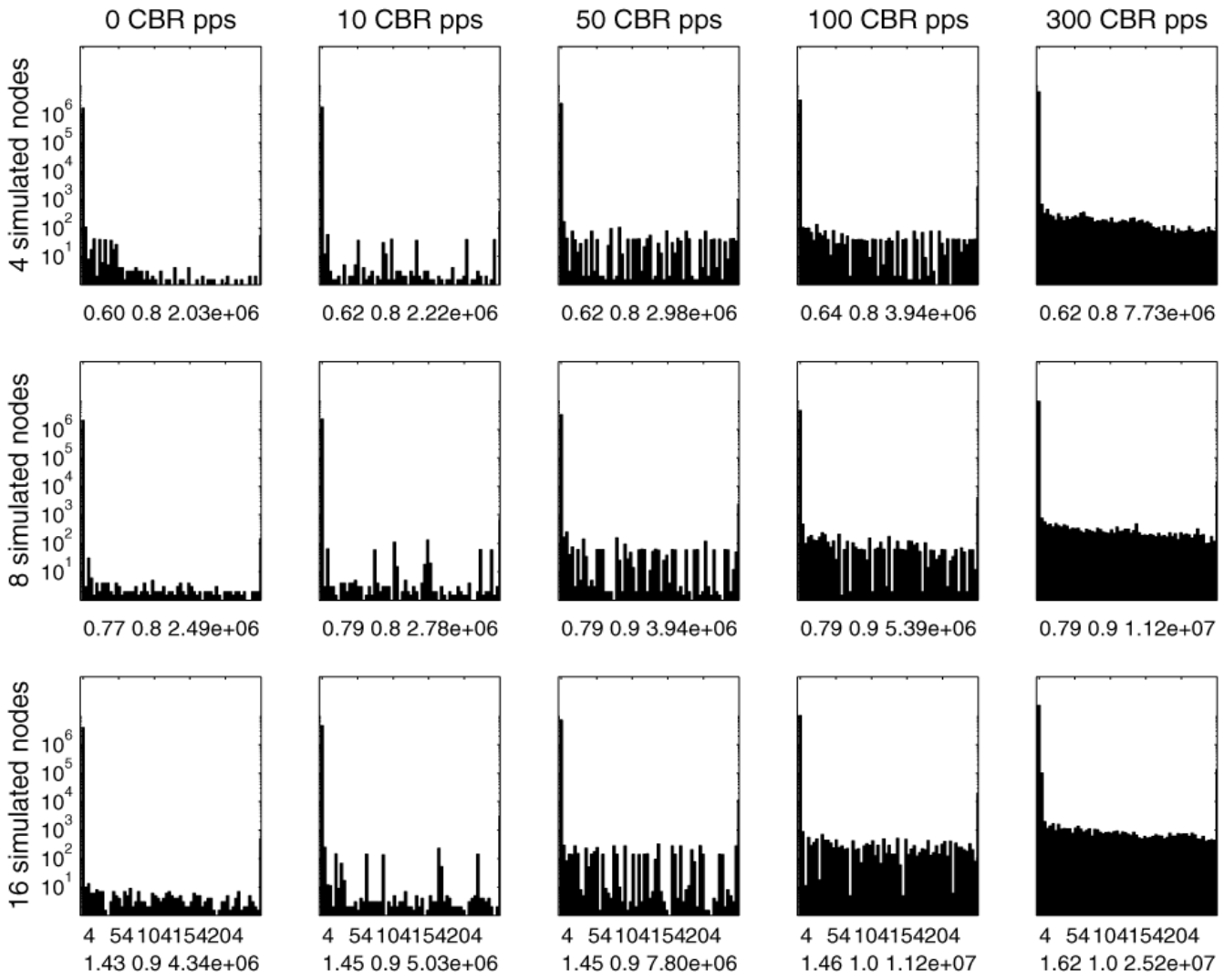
Experimental scenario:

- Conduct real FTP transfer between 2 physical nodes
- ***Vary*** the number of other simulated nodes in ns-2
- ***Vary*** the amount of simulated background traffic
- Produce a histogram of real-time ***time-lag*** over all events processed:

  (time processed – scheduled time)



Original Time-Lag Histograms



Breaking the Real-Time Requirement

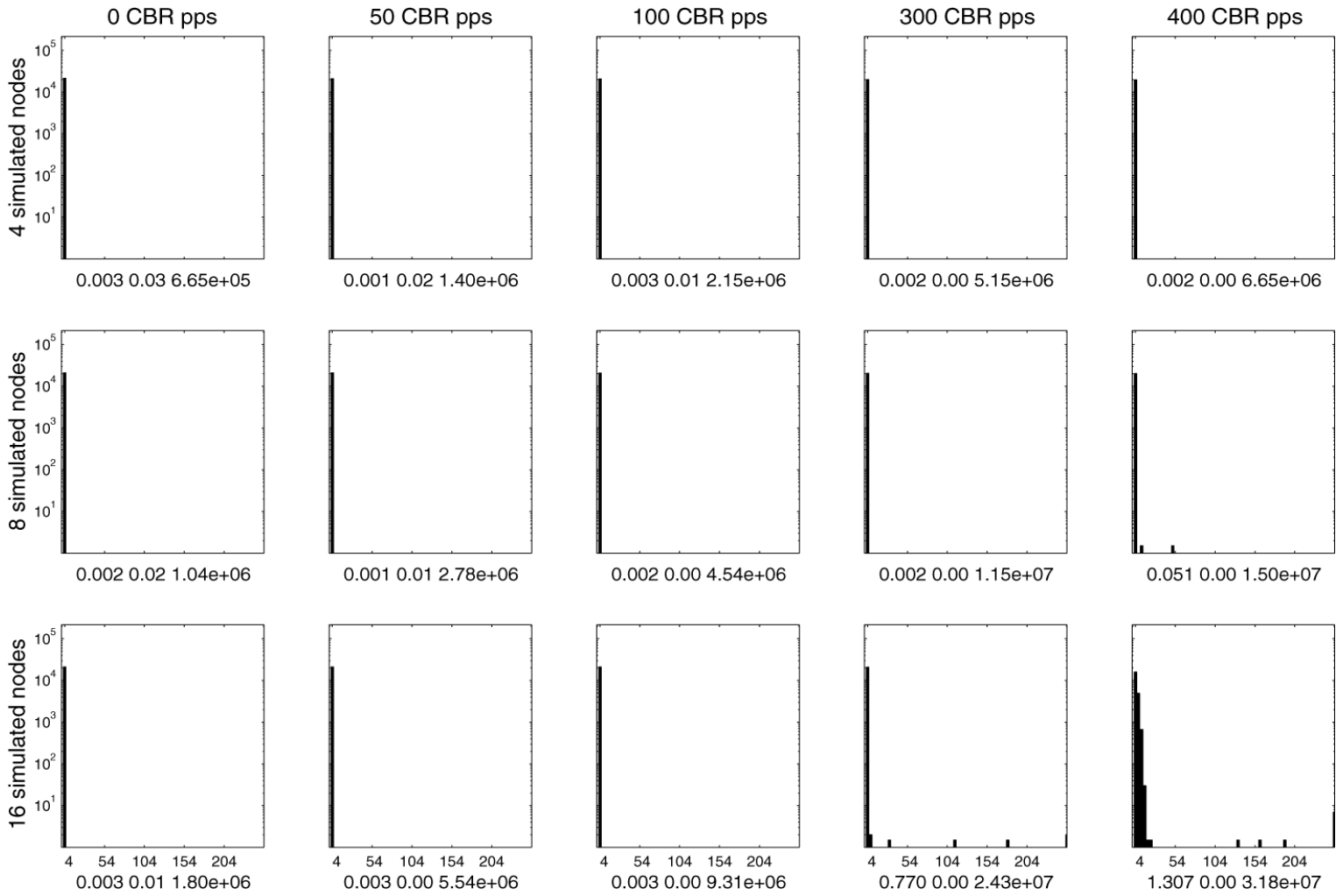
Two useful observations:

- ***Only time-lags of events at external nodes are visible:***
  - Allows more accurate characterization of scalability
  - Also, we want to execute pending ***visible*** events first, if we can guarantee not to violate causality
- ***The speed of light limits how fast causality can spread:***
  - Execute visible events ***before*** real-time by

$$\Delta t = \frac{\text{distance from nearest neighbor}}{\text{speed of light}}$$

  - During Δt, bursts of simulated events can happen
  - Order simulator event queue by (scheduled time – Δt)
  - We use constant overhead to compute the distance from the nearest neighbor at any time

Time-Lag Histograms after Optimization



Plans for Rest of FY00

Extend DSR to support adaptive QoS in ad hoc networks:

- Initial designs of path-state maintenance mechanisms complete
- Complete simulation evaluation and implementation in FreeBSD
- Plan DSR QoS demonstration at next GloMo PI meeting

Efficient multicast routing in DSR:

- Planning to maintain DSR's entirely on-demand behavior
- Initial design of most of protocol is complete
- Complete design and simulation evaluation, and implement if time

Enhance initial work on emulation of ad hoc networks:

- Work will include additional optimizations and improvements for scalability
- Also performance evaluation and validation of emulation results
- Use for initial performance evaluation of Coda on ad hoc networks

End date for Mobile Data Access contract is 30 September 2000